PATENT

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Applicants	:	Charles Averty et al.
Application No.	:	10/795,962
Filed	:	March 8, 2004
For	:	DEVICE AND PROCESS FOR USE IN ENCODING AUDIO DATA

Examiner	:	Brian Louis Albertalli
Art Unit	:	2626
Docket No.	:	851663.464
Date	:	August 22, 2008

Mail Stop Amendment
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

DECLARATION UNDER 37 CFR 1.131

I, Yao Xue, do hereby declare that:

1.   I am one of the inventors of the invention described and claimed in U.S. Patent Application Serial Number 10/795,962.

2.   The present application was filed on March 8, 2004, and claims priority to Singapore Application No. 200301300-0, which was filed on March 7, 2003.

3.   I have reviewed the Office Action mailed March 27, 2008, which rejected claims 1-4, 6, 7, 9-13 and 15-20 based in whole or in part on Chan, et al. ("A Low-Complexity, High Quality, 64-Kbps Audio Codec With Efficient Bit Allocation"), which was published in 2003.

4.   Exhibit A attached hereto is a true and correct copy of a portion of a document entitled "An Efficient Technique for MPEG1L2 Audio Psychoacoustic Masking." I am one of the authors of Exhibit A. Exhibit A bears a date and I have seen the date on the document. The date is prior to January 1, 2003. The date and other information have been

removed from the copy of the document submitted herewith, which I understand is permissible under Patent Office practice. Exhibit A describes an actual embodiment of a proposed encoder that was built and tested prior to the date on Exhibit A. The activities described in Exhibit A occurred in a WTO member country.

5.   Exhibit B attached hereto is a true and correct copy of a portion of a document entitled "A148: MPEG1 Layer2 Encoder Implementation Detailed Specifications Document." I am one of the authors of Exhibit B. Exhibit B bears a date and I have seen the date on the document. The date is prior to January 1, 2003. The date and other information have been removed from the copy of the document submitted herewith, which I understand is permissible under Patent Office practice. Exhibit B describes in more detail the configuration and results of testing the embodiment of an encoder described in Exhibit A. The activities described in Exhibit B occurred in a WTO member country.

6.   Exhibit C attached hereto is a true and correct copy of a portion of a document entitled "Precision Study on the MPEG-1 Layer-2 Encoder." I am one of the authors of Exhibit C. Exhibit C bears a date and I have seen the date on the document. The date is prior to January 1, 2003. The date and other information have been removed from the copy of the document submitted herewith, which I understand is permissible under Patent Office practice. Exhibit C describes in more detail the configuration and results of testing of the embodiment of an encoder described in Exhibit A. Pages 7-10 of Exhibit C illustrate some of the testing results in graphical form. The activities described in Exhibit C occurred in a WTO member country.

7.   Prior to January 1, 2003, I, along with the other applicants, reduced to practice the method of claim 1, a "mask generation process for use in encoding audio data, including: generating linear masking components from said audio data; generating logarithmic masking components from said linear masking components; and generating a global masking threshold from the logarithmic masking components." Exhibits A-C show the reduction to practice of claim 1. See, for example, Exhibit A at 1-6; Exhibit B at 5-6; Exhibit C at 4.

8.   Prior to January 1, 2003, I, along with the other applicants, reduced to practice the method of claim 2, the "mask generation process as claimed in claim 1 wherein said step of generating linear masking components includes: generating linear components in a

2

frequency domain from said audio data; selecting a first subset of said linear components as linear tonal components; and selecting a second subset of said linear components as linear non-tonal components." Exhibits A-C show the reduction to practice of claim 2. See, for example, Exhibit A at 2-3; Exhibit B at 5-6; Exhibit C at 2-4.

9. Prior to January 1, 2003, I, along with the other applicants, reduced to practice the method of claim 3, the "mask generation process as claimed in claim 2, including generating sound pressure levels from said linear components using a second-order Taylor expansion of a logarithmic function." Exhibits A-C show the reduction to practice of claim 3. See, for example, Exhibit A at 2; Exhibit B at 5-6; Exhibit C at 2-4.

10. Prior to January 1, 2003, I, along with the other applicants, reduced to practice the method of claim 4, the "mask generation process as claimed in claim 3, including generating a normalized value corresponding to an argument of said logarithmic function, and using said normalized value in said Taylor expansion." Exhibits A-C show the reduction to practice of claim 4. See, for example, Exhibit A at 2; Exhibit B at 5-6; Exhibit C at 2-4.

11. Prior to January 1, 2003, I, along with the other applicants, reduced to practice the method of claim 6, the "mask generation process as claimed in claim 2 wherein said step of generating a global masking threshold includes: decimating said linear tonal components and said linear non-tonal components; and generating masking thresholds from the decimated linear tonal components and the decimated linear non-tonal components." Exhibits A-C show the reduction to practice of claim 6. See, for example, Exhibit A at 3-4; Exhibit B at 5-6; Exhibit C at 2-4.

12. Prior to January 1, 2003, I, along with the other applicants, reduced to practice the method of claim 7, the "mask generation process as claimed in claim 6, wherein said step of generating a global masking threshold includes determining maximum components of said masking thresholds and predetermined threshold values." Exhibit A shows the reduction to practice of claim 7. See, for example, Exhibit A at 5-6.

13. Prior to January 1, 2003, I, along with the other applicants, reduced to practice the method of claim 9, the "mask generation process as claimed in claim 1 wherein said logarithmic masking components are generated using a second-order Taylor expansion of a

3

logarithmic function." Exhibits A-C show the reduction to practice of claim 9. See, for example, Exhibit A at 2; Exhibit B at 5-6; Exhibit C at 2-4.

14. Prior to January 1, 2003, I, along with the other applicants, reduced to practice the method of claim 10, the "mask generation process as claimed in claim 1, including generating masking thresholds from said logarithmic masking components using a masking function of the form:

$$vf = -17 * dz, \ 0 \le dz < 8 \text{."}$$

Exhibit A shows the reduction to practice of claim 10. See, for example, Exhibit A at 5.

15. Prior to January 1, 2003, I, along with the other applicants, reduced to practice the method of claim 11, the "mask generation process as claimed in claim 1 wherein said linear masking components include linear energy components, and said logarithmic masking components include logarithmic power components." Exhibits A-C show the reduction to practice of claim 11. See, for example, Exhibit A at 1-6; Exhibit B at 6; Exhibit C at 4.

16. Prior to January 1, 2003, I, along with the other applicants, reduced to practice the method of claim 12, the "mask generation process as claimed in claim 1 wherein said process is an MPEG-1 layer 2 audio encoding process." Exhibits A-C show the reduction to practice of claim 12. See, for example, Exhibit A at 1; Exhibit B at 3; Exhibit C at 3.

17. Prior to January 1, 2003, I, along with the other applicants, reduced to practice the method of claim 13, a "mask generation process for use in encoding audio data, including: generating logarithmic masking components; and generating respective masking thresholds from the logarithmic masking components using a masking function of the form:

$$vf = -17 * dz, \ 0 \le dz < 8 \text{."}$$

Exhibit A shows the reduction to practice of claim 13. See, for example, Exhibit A at 1-6, particularly at 5.

18.   Prior to January 1, 2003, I, along with the other applicants, reduced to practice the mask generator of claim 15, a "mask generator for use in encoding audio data, comprising: means for generating logarithmic masking components; and means for generating respective masking thresholds from the logarithmic masking components using a masking function of the form:

$$vf = -17 * dz, \ 0 \le dz < 8."$$

Exhibit A shows the reduction to practice of claim 15. See, for example, Exhibit A at 1-6, particularly at 5.

19.   Prior to January 1, 2003, I, along with the other applicants, reduced to practice the audio encoder of claim 16, an "audio encoder, comprising: means for generating linear masking components from said audio data; means for generating logarithmic masking components from said linear masking components; and means for generating a global masking threshold from the logarithmic masking components." Exhibits A-C show the reduction to practice of claim 16. See, for example, Exhibit A at 1-6; Exhibit B at 5-6; Exhibit C at 4.

20.   Prior to January 1, 2003, I, along with the other applicants, reduced to practice the computer readable storage medium of claim 17, a "computer readable storage medium having stored thereon program code that, when loaded into a computer, causes the computer to execute steps comprising: generating linear masking components from said audio data; generating logarithmic masking components from said linear masking components; and generating a global masking threshold from the logarithmic masking components." Exhibits A-C show the reduction to practice of claim 17. See, for example, Exhibit A at 1-6; Exhibit B at 3-6; Exhibit C at 3-4.

21.   Prior to January 1, 2003, I, along with the other applicants, reduced to practice the mask generator of claim 18, a "mask generator for an audio encoder, said mask generator comprising: means for generating linear masking components from input audio data; means for generating logarithmic masking components from said linear masking components; and means for generating a global masking threshold from the logarithmic masking

5

components." Exhibits A-C show the reduction to practice of claim 18. See, for example, Exhibit A at 1-6; Exhibit B at 5-6; Exhibit C at 4.
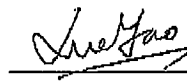
22.   Prior to January 1, 2003, I, along with the other applicants, reduced to practice the method of claim 19, a "psychoacoustic masking process for use in an audio encoder, comprising: generating energy values from Fourier transformed audio data; determining sound pressure level values from said energy values; selecting tonal and non-tonal masking components on the basis of said energy values; generating power values from said energy values; generating masking thresholds on the basis of said masking components and said power values; and generating signal to mask ratios for a quantizier on the basis of said sound pressure level values and said masking thresholds." Exhibit A shows the reduction to practice of claim 19. See, for example, Exhibit A at 1-6.

23.   Prior to January 1, 2003, I, along with the other applicants, reduced to practice the method of claim 20, an "MPEG-1-L2 encoder, comprising: means for generating energy values from Fourier transformed audio data; means for determining sound pressure level values from said energy values; means for selecting tonal and non-tonal masking components on the basis of said energy values; means for generating power values from said energy values; means for generating masking thresholds on the basis of said masking components and said power values; and means for generating signal to mask ratios for a quantizier on the basis of said sound pressure level values and said masking thresholds." Exhibit A shows the reduction to practice of claim 20. See, for example, Exhibit A at 1-6.

6

24.   I further declare that all statements made herein of my own knowledge are true and that all statements made on information and belief are believed to be true; and further that these statements were made with the knowledge that willful false statements and the like so made are punishable by fine or imprisonment, or both, under Section 1001 of Title 18 of the United States Code and that such willful false statements may jeopardize the validity of the application or any patent issued thereon.

DATE: _Aug 22. 08_

Yao Xue

1207710_1.DOC

1 Means Of Solving The Problems

The PAM1 used in MPEG1-layer2 audio encoder has been analyzed in detail to remove the repetitive power/energy conversion, and to simplify the complicated functions for the global masking threshold calculation. Thereby, computational complexity is reduced without sacrificing quality excessively.

2 Operations Of The Invention

The PAM can be split into 8 steps. These steps will be further discussed in the next section. Of these steps there are some which involve transformations from power values to energy values, or vise versa, by the following formulae:

$$Power = 10 * \log 10(energy)dB$$

$$Energy = 10^{0.1 * Power}$$

From a real-time implementation point of view, these successive conversions not only require extra processing due to the representation of the logarithm and the power functions by Taylor series expansions, but also incur a repeated loss in precision.

Therefore the ISO algorithm has been revised so that there is only one conversion which is from power to energy domain. This conversion is actually a calculation of common logarithm, which is implemented by using second-order Taylor expansion.

Inside the PAM, the most computational intensive step is the calculation of global masking threshold. The global masking threshold $LT_g(i)$ is found by summing the powers corresponding to the individual masking thresholds and the threshold in quiet. Again there are conversions from power to energy and back to power. These conversions are replaced by comparison operation, which requires much less computational power. The calculations of individual masking thresholds involves the use of complicated masking functions characterized by different slopes. They are simplified according to the significance of their different masking abilities.

3 Detail Description Of Preferred Embodiments

The rest of this patent is presented in the following manner. Firstly, a step-by-step description of the PAM1 is provided. This serves to explain the overall process and describe the significance of each step in the whole model.

Inside each step-by-step description, the proposed modifications is elaborated. These modifications are aimed at maximally reducing the computational load while keeping the effect on the quality to the minimum.

Step 1: FFT Analysis

The time-to-frequency mapping is done by a 1024-point FFT. The FFT is calculated directly from the input PCM signals, windowed by a Hann window.

After the FFT analysis, an array of 512 energy values are calculated from the 1024 FFT output. These energy values are henceforth converted to PSD (power spectral density) values. The energy values are later discarded. A normalization to the reference level of 96 dB SPL (Sound Pressure Level) is made in such a way that the maximum PSD value corresponds to 96 dB.

Proposed Modification:

The energy values are not converted to PSD values. There is also no normalization for these values.

Step 2: Determination of SPL

The SPL value L_{sb} in subband n is computed by:

$$L_{sb}(n) = MAX \lfloor X_{spl}(n), 20*\log(scf_{max}(n)*32768)-10 \rfloor \; dB$$

with $X_{spl(n)} = 10*\log_{10} \left(\sum_k 10^{X(k)/10} \right) dB$ (k

in subband n)

where $X(k)$ is the PSD value of index k, and $scf_{max}(n)$ is the maximum of the three scalefactors of subband n within a frame. The "-10 dB" term corrects for the difference between peak and RMS level. The SPL value L_{sb} is computed for every subband n.

Proposed Modification:

L_{sb} is computed by the same formulae with,

$$X_{spl(n)} = 10*\log_{10} \left(\sum_k X(k) \right) + 96 \; dB \qquad\qquad (k \text{ in subband n})$$

since $X(k)$ is now in the energy domain. The "96 dB" term is used for normalization.

The operation of common logarithm is approximated using a second order Taylor expansion. Assuming `Ipt` is the input. First, normalize `Ipt` so that $Ipt = (1-x)2^m$, where $0.5 < 1-x \leq 1$. Using second order Taylor expansion,

$$\ln(1-x) \approx -x - x^2/2$$

the result becomes,

$$\log_{10} Ipt = [m*\ln 2 - (x + x^2/2)]*\ln 10$$

Step 3: Locating the tonal components

The tonality of a masking component has an influence on the masking threshold. For calculating the global masking threshold, it is necessary to derive the tonal and non-tonal components from the FFT spectrum.

This step starts with the determination of local maxima. A spectral line $X(k)$ is labeled as a local maximum if $X(k) > X(k-1)$ and $X(k) \geq X(k+1)$. The local maximum is then put in the list of tonal components if $X(k) - X(k+j) \geq 7dB$, where j is the searching range varies with different k.

If $X(k)$ is found to be a tonal component, then its value is updated by

$$X_{tonal}(k) = 10 * \log_{10}\left(10^{X(k-1)/10} + 10^{X(k)/10} + 10^{X(k)/10}\right)$$

Next, all spectral lines within the examined frequency range are set to $-\infty$ dB.

Proposed Modification:

A local maximum $X(k)$ is listed as tonal component if $X(k)*10^{-0.7} \geq X(k+j)$.
The updated value of tonal component $X_{tonal}(k)$ is simply $X(k-1) + X(k) + X(k+1)$

Step 4: locating the non-tonal components

This step calculates the intensity of the non-tonal components within the bandwidth of critical bands, which varies with the center frequency of the specific critical band. 26 critical bands are used for 48 kHz sampling rate.

The non-tonal (noise) components are calculated from the spectral lines remaining after the tonal components are zeroed. Within each critical band, the power of the remaining spectral lines are summed to form the SPL of the new non-tonal component $X_{noise}(k)$ corresponding to that critical band. The number k is the index number of the spectral line nearest to the geometric mean of the critical band.

Proposed Modification:

$X_{noise}(k)$ is produced by summing the energy of the remaining spectral lines. Hence there is only plain addition instead of logarithmic addition, which needs to convert power values to energy ones and then convert the summation back to the power domain.

Step 5: Decimation of tonal and non-tonal components

Decimation is a procedure that is used to reduce the number of maskers which are considered for the calculation of the global masking threshold.

Tonal components $X_{tonal}(k)$ and non-tonal components $X_{noise}(k)$ are considered for the calculation of the masking threshold only if,

$$X_{tonal}(k) \geq LT_q(k) \text{ or } X_{noise}(k) \geq LT_q(k)$$

where $LT_q(k)$ is the absolute threshold (or threshold in quiet) at the frequency of index k. These values are in tables given by MPEG.

Decimation is carried out on two or more tonal components within a distance of less than 0.5 Bark[*]. The component with the highest power is kept while the smaller component(s) are removed from the list of tonal components. For this operation, a sliding window in the critical band domain is used with a width of 0.5 Bark.

Proposed Modification:

- A new table $LT_q E(k)$ is generated from the table of absolute threshold $LT_q(k)$ by,

$$LT_q E(k) = 10^{\log_{10}[LT_q(k) - 96]/10}$$

The "-96" term is used for the denormalization.

Afterwards the tonal and non-tonal components $X_{tonal}(k)$ and $X_{noise}(k)$ are compared against $LT_q E(k)$ and the ones which are smaller are winnowed out.

Up till this step the spectral lines $X(k)$ remain in the energy domain. At the end of this step when the decimation process is over, all the spectral lines are converted into the power domain by using the common logarithmic formula described in *Step 2*. This conversion is followed by a normalization to the reference level of 96 dB.

Step 6: Calculation if individual masking threshold

This step calculates the individual masking thresholds and is a precursor of calculating the global masking threshold. Of the original 512 spectral lines, indexed by k, only a subset, indexed by i, is considered for the global masking threshold calculation. The number of lines n in the subsampled frequency domain is different depending on the sampling rate. For 48 kHz sampling rate, $n = 126$.

Every tonal and non-tonal component is assigned an index i which is most closely corresponds to the frequency of the original spectral line $X(k)$.

The individual masking thresholds of both tonal and non-tonal components, LT_{tonal} and LT_{noise}, are given by the following expressions:

---

[*] Bark scale is a frequency scale on which the frequency resolution of the ear is approximately constant. See Zwicker"Subdivision of the Audible Frequency Range into Critical Bands", J. Acoustical Society of America, vol. 33, p. 248, February 1961.

$$LT_{tonal}[z(j),z(i)] = X_{tonal}[z(j)] + av_{tonal}[z(j)] + vf[z(j),z(i)] \ dB$$

$$LT_{noise}[z(j),z(i)] = X_{noise}[z(j)] + av_{noise}[z(j)] + vf[z(j),z(i)] \ dB$$

In these expressions i is the index of the spectral line at which the masking threshold is calculated while j is that of the masker. $z(i)$ is the Bark scale value of the ith spectral line while $z(j)$ is that of the jth line. The term $X[z(j)]$ is the SPL of the masker (tonal or non-tonal). The term av, called masking index, is given by:

$$av_{tonal} = -1.525 - 0.275 * z(j) - 4.5 \ dB$$

$$av_{noise} = -1.525 - 0.175 * z(j) - 0.5 \ dB$$

vf is the masking function of the masker and is characterized by different lower and upper slopes, which depends on the distance in Bark scale dz, $dz = z(i) - z(j)$. The masking function, which is the same for tonal and non-tonal maskers, is given by:

$$vf = 17 * (dz + 1) - 0.4 * X[z(j)] - 6 \ dB, \text{ for } -3 \le dz < -1 \ Bark$$

$$vf = \{0.4 * X[z(j)] + 6\} * dz \ dB, \text{ for } -1 \le dz < 0 \ Bark$$

$$vf = -17 * dz \ dB, \text{ for } 0 \le dz < 1 \ Bark$$

$$vf = -17 * dz + 0.15 * X[z(j)] * (dz - 1) \ dB, \text{ for } 1 \le dz < 8 \ Bark$$

In these expressions $X[z(j)]$ is the SPL of the masker. For reasons of implementation complexity, the masking is no longer considered if dz < -3 Bark or dz ≥ 8 Bark.

Proposed Modification:

The calculation of the masking function vf is the most computationally intensive part within this step. This masking function can be categorized into two types: downward masking (when $dz < 0$) and upward masking (when $dz \ge 0$). The existing research shows that the downward masking is considerably less significant than upward masking [3]. Therefore in our implementation, only the upward masking is considered.

Further analysis on the last expression of vf indicates that the second term, $0.15 * X[z(j)] * (dz - 1)$, is normally one tenth of the first term, $-17 * dz$. Hence the second term can be safely discarded.

As a result, the masking function vf is calculated by a single expression $vf = -17 * dz$ ($0 \le dz < 8$). This simplification greatly reduces the computational load.

The masking index av is left unchanged since it has a significant contribution to the individual masking threshold LT and does not consume much computational power.

Step 7: Calculation of the global masking threshold

The global masking threshold $LT_g(i)$ at the i'th frequency sample is found by summing the powers corresponding to the individual masking thresholds and the threshold in quiet.

$$LT_g(i) = 10\log_{10}\left[10^{LT_q(i)/10} + \sum_{j=1}^{m} 10^{LT_{tonal}[z(j),z(i)]}/10 + \sum_{j=1}^{n} 10^{LT_{noise}[z(j),z(i)]}/10\right]$$

The total number of tonal maskers is given by m, and the total number of non-tonal maskers is given by n. The threshold in quiet LT_q is revised by an offset of $-12\,dB$ for bit rates ≥ 96 kbps per channel.

Proposed Modification:

Again the summation of powers here is wasting lots of computing power. Therefore the summation operation is replaced by comparison for simplicity at the expense of occasional over allocation. This new approach can be expressed as:

$$LT_g(i) = \max\left[LT_q(i) + \max_{j=1}^{m}\{LT_{tonal}[z(j),z(i)]\} + \max_{j=1}^{n}\{LT_{noise}[z(j),z(i)]\}\right]$$

The maxima of m tonal maskers and n non-tonal maskers are identified. They are then compared with $LT_q(i)$. The maximum of these three values is selected as the global masking threshold at the i'th frequency sample.

Step 8: Calculation of the signal-to-mask ratio

The minimum masking level $LT_{min}(n)$ in subband n is determined by the following expression:

$$LT_{min}(n) = Min\left[LT_g(i)\right] \, dB, \text{ for } f(i) \text{ in subband } n$$

where $f(i)$ is the frequency lines inside subband n. A minimum masking level $LT_{min}(n)$ is computed for every subband.

Finally the signal-to-mask ratio for every subband n is computed by subtracting the minimum masking level from the SPL value of that subband,

$$SMR_{sb}(n) = L_{sb}(n) - LT_{min}(n)$$

Proposed Modification:

There is no modification for this step.

DSP R&D Centre

Detailed Specifications Document
A148/D0070/1.0

1 Introduction

1.1 Purpose

This document intends to present the details of the project A148, and how the results were obtained. Starting from the floating-point implementation, the code was first cleaned up. Later, a few significant changes were made to the algorithm, and then the fixed-point implementation was realized before it was cross-compiled for the MMDSP+. Finally, memory and MCPS optimizations completed the project.

1.2 Scope

This document should serve as a guide for further optimizations of the algorithm as well as for using it in applications.

1.3 References

[1]   International Standard: ISO/IEC 11172-3 "Information technology – Coding of moving pictures and associated audio for digital storage media at up to about 1.5Mbps".

[2]   "Precision study on MPEG1-Layer2 encoder", *project document [D0066]*

[3]   User Manual and Style Guide for the MMDSP+ *compiler*

[4]   User Manual for MSIM *profiling*

[5]   MMDSP+ DSP core

[6]   "General guide to implement logarithmic and exponential operations on a fixed-point DSP", *application report*, Texas Instruments

[7]   OPERA user manual, version 3.0

[8]   "Test Plan Document for MPEG1-Layer2 encoder", *project document [D0063]*

[9]   "Requirement Specification for MPEG1-Layer2 encoder", *project document [D0059]*

1.4 Definitions, Acronyms, and Abbreviations

DAC:	Digital to Analog Signal Converter
ISO:	International Standard Organization
DSP:	Digital Signal Processor
DVD:	Digital Video Disc
FFT:	Fast Fourier Transform
ROM:	Read Only Memory
RAM:	Read Access Memory
MCPS:	Million Cycles Per Second
ALU:	Arithmetic Logic Unit
DDCE:	Dolby Digital Consumer Encoder
ITU:	International Telecommunication Union
PEAQ:	Perceptual Evaluation of Audio Quality
ODG:	Objective Difference Grade

2 Floating Point

2.1 Original Code

The original MPEG code is stored in one single directory. It contains both the encoder for layers 1 and 2, and decoder for all the three layers. Since only the encoder is implemented in this project, the source code files concerning the decoder (such as decode.c, musicout.c, huffman.c, decoder.h, huffman.h) are removed.

The files used for encoder are:
- musicin.c   :   input parameters, encoding process and bitstream generation
- encode.c   :   encoder routines
- tonal.c   :   psychoacoustic model I
- psy.c   :   psychoacoustic model II
- subs.c   :   FFT analysis for psychoacoustic model II
- common.c :   file containing common functions and format conversion routines
- encoder.h :   header file for encoder routines
- common.h :   header file for common information

The floating code from MPEG is not optimized at all. The purpose of various switches in the source code is not always clear.

2.1.1 Compilation

The MPEG example code provides a single Makefile with two sets of compilation flags for both Unix and DOS. To make the program in Unix, simply use the **make** command. To run the application, launch **musicin**. Various command line arguments can be parsed by the program.

2.1.2 Limitations

The MPEG1-Layer2 Encoder has some limitations compared to other encoders such as DDCE. This MPEG encoder is of relatively low complexity. As a tradeoff, its performance is also limited.

In accordance with Philips DVD RAM requirements, only stereo signals with bit rates of 256kbps and 384kbps are supported. Mono and joint stereo modes are disabled.

For the bit rate of 384 kbps, the encoder has a compression ratio of around 4.

2.2 Storage and version control

The project is stored in the public vobs storage of Clearcase mounted on **/vobs/a148**. The first level of subdirectories contains the code (**ccode**) and the documents (**documents**).

The original code from ISO is stored under the label **ISO4.4_March_96**.

2.3 Encoder Specification

Starting with this ISO code, the first task was to extract a clean version, which would satisfy our specifications for the MPEG encoder. In line with this purpose, preliminary changes (mainly extraction and deletion) were made in order to come up with the basic skeleton code for our customized MPEG encoder.

In order to better understand those changes, it is beneficial to introduce our specifications for the MPEG encoder at this point, which are as follows:

- The code is built for the Unix environment.

- The encoder accepts only raw PCM inputs.

- The input, processing and output is on a per-frame basis (the ISO code was not in accordance with this, as the output was not done on a per-frame basis).

- The only sampling frequency used would be 48kHz (Optimization of tables is achieved, tables required for 32kHz and 44.1kHz are ignored).

- The only two bit-rates used would be 256 kbps and 384 kbps (the other bit-rates as specified by a 4-bit *bitrate-index* would be ignored).

- Psychoacoustic model 1 would be used.

2.4 Preliminary changes

2.4.1 Redundancy

First of all, since only layer 2 is to be implemented, all the functions and tables used only in layer 1 are removed.

In several locations of the ISO code, redundancy is present for more flexibility in the design stage. For example, there are functions dealing with `aiff` input. There are also switches such as MACINTOSH, MSC60, and CONVEX, to accommodate various operating systems. These functions, switches and related codes were removed.

The original code uses structures extensively, which is inefficient for fast implementation. Therefore, efforts were taken to remove most of the structures, such as `g_thres`, `g_ptr`, `mask` and `mask_ptr`. The only ones left are those for header and bit stream information.

In the ISO code, most of the tables required for psychoacoustic analysis are read in from files. Differentiation was also made based on layer and sampling frequency. Since our requirement are Layer 2 and sampling frequency of 48 kHz, tables necessary for this configuration were extracted and put into arrays to minimize memory usage and to simplify file processing.; the remaining files were removed.

There are two psychoacoustic models available in the original code. Model 2 is more complex but has better performance for lower bit rates. However, after a

series of quality tests, model 1 was found to have better quality at the two required bit-rates.

Psychoacoustics Models
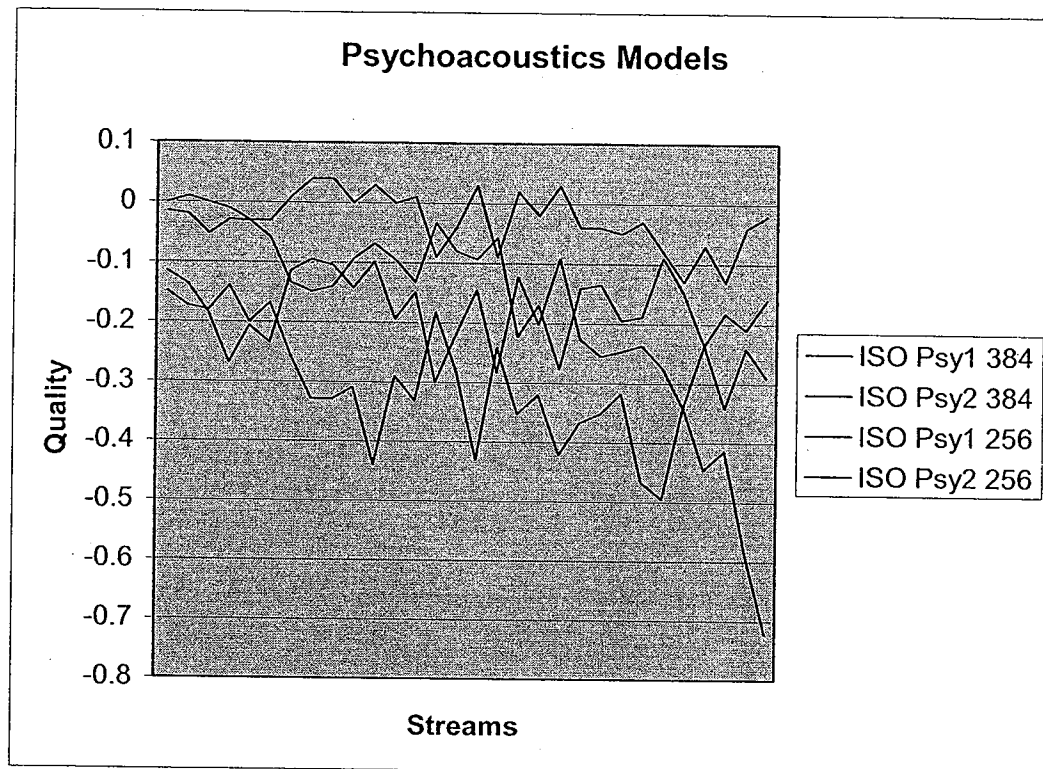
Therefore, everything pertaining only to model 2 was removed. This includes files **psy.c** and **subs.c**, some structures, and several functions in the remaining files.

Finally, to speed up the testing and execution, the function `obtain_parameters` was removed. A set of default parameters were specified. As a result, it is no longer compulsory to key in parameters during execution. However, the command line arguments can still be used for overriding any default parameter.

2.4.2 Compiling

It has been decided that the source codes for all versions of the application should be the same, and thus extensive use of preprocessor directive **#ifdef** should be used.

For better understanding and compatibility with the fixed point and mmdsp+ modes, the floating makefile with the name **Makefile** has been rewritten. The changes made in the original code for removing the redundancy are also reflected in the new **Makefile**. This **Makefile** is now for Unix only.

The variable used to identify the floating-point application is **FL** (to identify the portion of codes relevant only to the fixed point implementation the variable **BT** as in Bit True is used). Several switches need to be enabled in order to access the floating version of corresponding functions. These switches are: **WINDOWFLOAT,**

SCALEFLOAT, PICKSCALEFLOAT, FILTERFLOAT, HANNFLOAT, FFTFLOAT.

The object files and executables are generated by running **make**. The name of the executable is **musicin**. The command line arguments remain the same.

After the precision study, a new makefile **lfl.mak** was written. It is the same as **Makefile** except that another variable **LFL** is added to identify the part of codes where computing precision has been limited. The executable remains `musicin`, which is generated by running `make -f lfl.mak`.

2.5 Structural changes

Once the preliminary changes were done, and a neat code suited to our encoder specifications was ready, the next stage was to do a precision study of the code. The proceedings and results of this stage can be found in [2]. During this process, a few structural changes have been made.

2.5.1 Bit allocation table

The ISO code bit allocation tables are two-dimensional tables of different sizes for different subbands, and are stored in separate ASCII files. On closer observation, there are repetitions in these tables.

The idea then was to compress them into unique sets of two-dimensional tables (of different sizes). In addition, a table was created that contains the addresses of the unique tables according to which subband it refers. This simple technique resulted in an approximate nine times reduction in memory for the 48kHz sampling frequency. For a complete implementation of the MPEG encoder (inclusive of all sampling frequencies), this technique would save a lot more ROM.

2.5.2 Energy to power conversion

The psychoacoustic model 1 provides the signal-to-mask (SMR) ratio for all the 32 subbands for the purpose of bit allocation. For that, it is necessary to determine the maximum signal level and the minimum masking threshold for each subband. The minimum masking threshold is derived from an FFT of the input PCM signal, followed by a psychoacoustic model calculation.

It was observed that in the psychoacoustic model calculation, there are repeated power to energy and energy to power conversions, with power and energy defined as:

$$Power = 10 * \log 10(energy) dB \qquad (1)$$

$$Energy = 10^{0.1 * Power} \qquad (2)$$

From a real-time implementation point of view, these successive conversions not only require extra processing due to the representation of the logarithm and the power functions by Taylor series expansions, but also incur a repeated loss in precision.

Therefore, after a thorough analysis of the ISO code, algorithmic changes were made to eliminate the unnecessary conversions. Energy values were obtained from FFT and used throughout the psychoacoustic model calculation until the function for determining the masking threshold, where those values are converted into power values. For more details, please refer to [2].

2.5.3 Approximation formulae

In the fixed point implementation, the common logarithm, power of 10, and square root functions are all approximated by formulae obtained from Taylor expansions. The precision of these approximations is proportional to the order of expansions, but so is the computational intensity. Therefore, trade-offs have to be made. According to [6], a second order approximation meets the precision requirements for both common logarithm and power of 10 calculations.

2.5.3.1 Common logarithm

Assuming `Ipt` is the input. First, normalize `Ipt` so that $Ipt = (1 - x)2^m$, where $0.5 < 1 - x \leq 1$. Using second order Taylor expansion,

$$\ln(1 - x) \approx -x - x^2 / 2 \tag{3}$$

the result becomes,

$$\log_{10} Ipt = [m * \ln 2 - (x + x^2 / 2)] / \ln 10 \tag{4}$$

2.5.3.2 Power of 10

Assuming `Ipt` is the input in dB. In this MPEG code, $Ipt \leq 0$. First, normalize `Ipt` so that $Ipt / 10 = norm + x$, where $-0.5 < x \leq 0.5$. Using second order Taylor expansion for e^x,

$$e^x = 1 + x + x^2 / 2 \tag{5}$$

the result becomes,

$$10^{Ipt/10} = 10^{norm} * [1 + \ln 10 * x + (\ln 10 * x)^2 / 2] \tag{6}$$

2.5.3.3 Square root

The square root function in this MPEG code takes in an integer, `Ipt,` as the input and returns an integer as output. The precision requirement is low. As such, a second order approximation formula is used.

First, the input `Ipt` is normalized so that $Ipt = (1 - x)2^m$, where $0 < 1 - x \leq 1$. Using order-2 Taylor series,

$$\sqrt{1 - x} = 1 - 0.5 * x - 0.125 * x^2 \tag{7}$$

Therefore,

$$\sqrt{Ipt} = 2^{m/2}(1 - 0.5 * x - 0.125 * x^2) \tag{8}$$

for m even, or

$$\sqrt{Ipt} = 2^{(m-1)/2}(1 - 0.5 * x - 0.125 * x^2) * \sqrt{2} \tag{9}$$

for m odd.

EXHIBIT C

DSP R&D , Singapore

Technical Report/v1.0

Precision Study on the MPEG-1 Layer-2 Encoder

Abstract - *The MPEG-1 Layer-2 Encoder is part of DVD-RAM system. For its cost-effective implementation, several approaches had been investigated. A model with limited precision has been tested on real audio tracks to get closer to the optimal solution. This report presents the results of this precision study.*

	Name	**Designation**	**Signature**
Created by :	Xue Yao, Ranjot Singh	DSP systems Design Engineer	
Checked & Approved by :	Kay Das	Director, R & D Programmes	
Version :	1.0	**Date of Isst**	

1 Introduction

MPEG-1 Layer-2 encoder is a perceptual encoder which uses the frequency masking property of the human auditory system to compress 2-channel audio signals into bitstream for storage. The encoder algorithm is not standardized, and may use various means for encoding such as estimation of the auditory masking threshold, quantization, and scaling. However, the encoder output must be such that a decoder conforming to the ISO specifications will produce audio signals suitable for the intended application [1].

In the basic approach, input audio samples are fed into both an analysis filter bank and a psychoacoustic model. The filter bank creates a filtered and subsampled representation of the input audio stream. The psychoacoustic model creates a set of data to control the quantizer and coding. These data could be different, depending on the actual encoder implementation.

There are different encoding modes, such as stereo/joint stereo, sampling frequency of 32k/44.1k/48k etc. In our implementation, a fixed combination of 48k/stereo/psychoacoustic model 1 is used. The bitrate is set as either 384kbps or 256kbps.

The psychoacoustic model has the greatest computational requirements in the encoder. An efficient implementation of this block can significantly decrease the amount of computation and make real time operation of the encoder more attainable. Therefore, an algorithmic improvement has been made to the original encoder.

Following the algorithmic change, the influence of the precision of the data is studied. A model with limited precision has been tested on real audio tracks to get closer to the optimal solution.

2 Precision Study Details

The MPEG-1 Layer-2 encoder is to be implemented on the MMDSP+ DSP. The MMDSP+ is a 24 bit DSP processor. This implies that the precision we want to achieve is 24 bits ideal (single precision), and hence study if this is possible. This is what the precision study is all about. The entire exercise means range and precision study of the various tables, variables and collecting their precision requirements. This would give a better idea of where single, mixed or double precision would be required, and where a trade-off could be achieved for lower processing requirements but at the cost of precision.

During the process of precision study various changes, algorithmic and structural changes were made to the code, for the sake of implementation feasibility (in direct relation to processing and precision optimisation). One of the most critical changes was within the psycho-acoustic module.

For our implementation, the Psychoacoustic model 1 is chosen. This model provides the signal-to-mask (SMR) ratio for all the subbands for the purpose of bit allocation of the 32 subbands. For this purpose, it is necessary to determine for each subband, the maximum signal level and the minimum masking threshold. The minimum masking threshold is derived from an FFT of the input PCM signal, followed by a psychoacoustic model calculation.

Now the Psychoacoustic model calculation involves quite a few steps. Step 1 is a FFT analysis, wherein after the FFT analysis the ISO code does the energy calculation and henceforth the power spectral density (X(k)) calculation. Henceforth, the ISO code only saves the PSD values and the energy values are discarded. Step 2 is the determination of sound pressure level where the alternative sound pressure level corresponding to the nth subband is the sum of the spectral components in the subband. This, according to the procedure followed by the ISO code, would involve a transformation from the power values to energy values and then back to power values by the following formulae:

Power = 10 * log10(Energy) dB (1)

Energy = $10^{0.1*Power}$ (2)

After the third step of considering the threshold in quiet, step 4 involves the finding of tonal and non tonal components. To make lists of the spectral lines X(k) that are tonal or non-tonal, the following three operations are performed:

a) Labelling of local maxima which involves comparisons between the spectral values

b) Listing of tonal components and calculation of the sound pressure level

c) Listing of non-tonal components and calculation of the power

For step(b) again the calculation of the sound pressure level involves the summation of the X(k-1), X(k) and X(k+1) spectral components and therefore a power to energy and back to power conversion as far as the ISO code is concerned. Step 5 is the decimation of tonal and non-tonal components, which basically involves comparison of the components with reference values stored in tables.

Thus after a thorough analysis of the ISO code, it was observed that from a fixed point implementation point of view, these repeated power to energy and energy to power conversion would not only mean extra processing due to the representation of the logarithm and the power (of 10) functions by Taylor series expansions, but would also mean a repeated loss in precision which indirectly would relate to computational intensity again as the order of the expansions would decide the precision.

It was also observed that this energy to power conversion done in Step 1 after the FFT analysis could be procrastinated to Step 6, which was the calculation of the individual masking thresholds. The individual masking thresholds of both tonal and no-tonal components are given by certain standard expressions which are in dB form. After some mathematical analysis it was realized that an analogous energy expression would do more harm than good as it increased the complexity for one of the expressions.

Thus it was decided that the energy to power conversion be delayed to the sixth stage, wherein the energy would be finally converted to the power form so as to fit into the expressions for the calculation of the individual masking thresholds of the tonal and non-tonal components. Accordingly the ISO code was changed to accommodate the above statement. All comparisons were done with energy values and reference dB values were converted to corresponding energy values so as to be consistent with the comparisons.

Another important point noticed during the precision study was in the scalefactor calculation. The calculation of the scalefactor for each subband is performed every 12 subband samples. The maximum of the absolute value of these 12 samples is determined. The lowest value in the Layer II scalefactors table, provided in the standard, which is larger than this is used as the scalefactor. This scalefactor table has a requirement of more than 24 bit precision, and henceforth an attempt to do the scalefactor calculation with the subband samples and the table values in 24 bit resulted in a large precision loss, as wrong scalefactors were chosen. Therefore 48 bit precision was chosen for the scalefactor table and an array of 32 subband samples for the calculation of this module, and the resultant precision loss from this module was nil.

3 Test and Results

Constant testing performed during the precision study, involved an original PCM music sample being encoded-decoded and then being compared with the original PCM music sample. A point to be noted here is that a 480 sample delay was observed between the original and the encoded-decoded PCM samples. This delay was therefore compensated for during the testing.

Some sample final results are as follows (for original PCM sample = BueanasAires.pcm):

Average number of bits in error between original ISO PCM(original PCM sample encoding with original ISO code and then standard decoding) and the original PCM : 6.505020 bits
Worst case of bits in error : 12 bits
Highest Probability of bits in error : 6 bits

Average number of bits in error between modified ISO PCM(original PCM sample encoding with modified ISO code and then standard decoding) and the original PCM : 6.406473 bits
Worst case of bits in error : 12 bits
Highest Probability of bits in error : 6 bits

The above testing was also performed on a large number of audio tracks involving percussion instruments, piano, harp, saxophone, castanet, voice, choir, etc. The results of the comparison of the original ISO and the modified ISO individually with the original PCM were then compared. Some of these results are as follows:

kundun

drouet7

celin
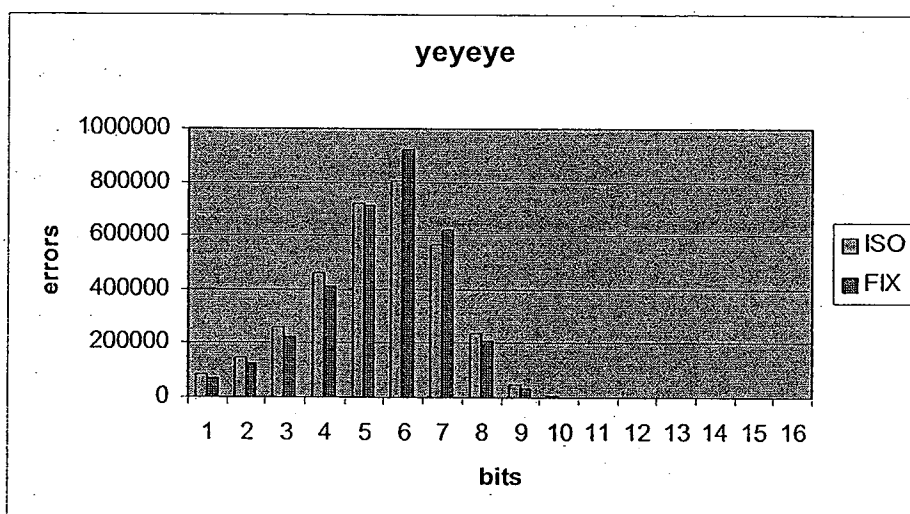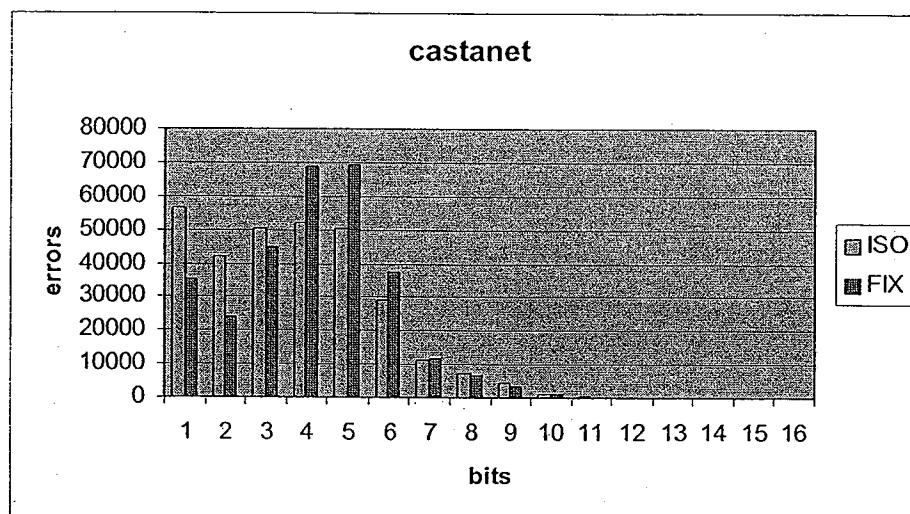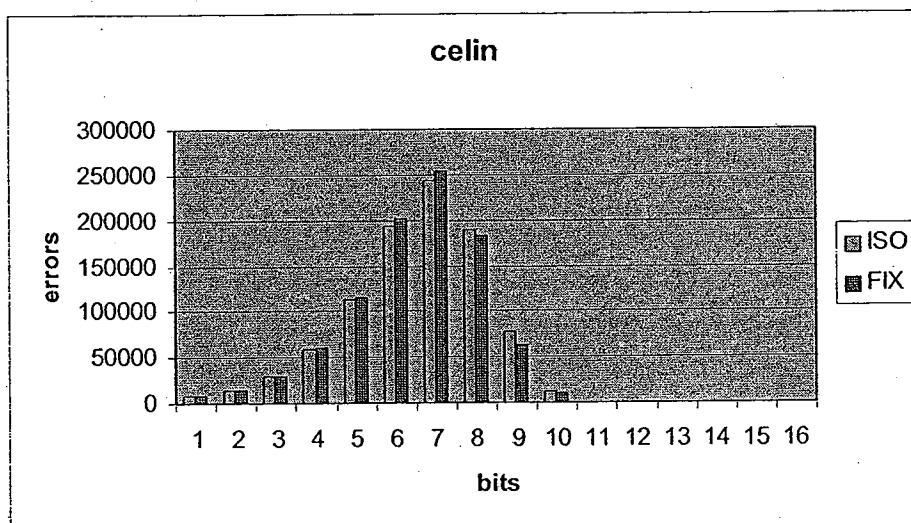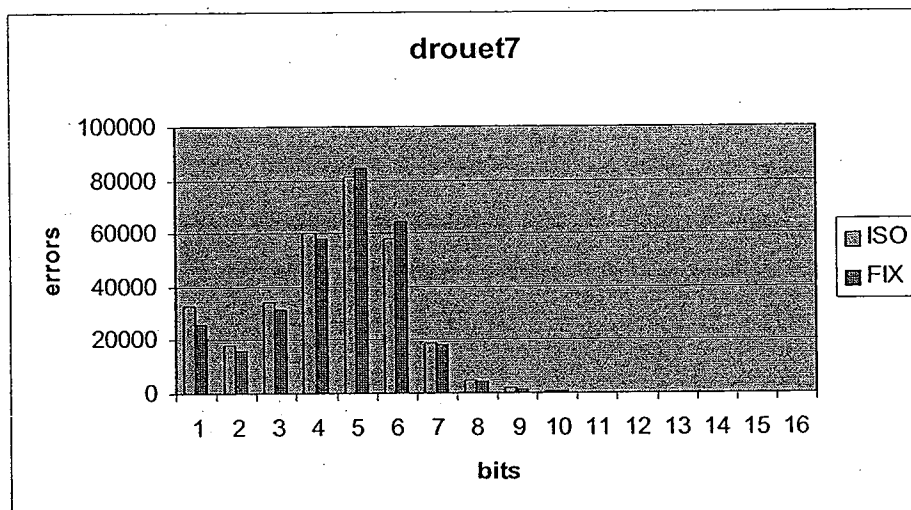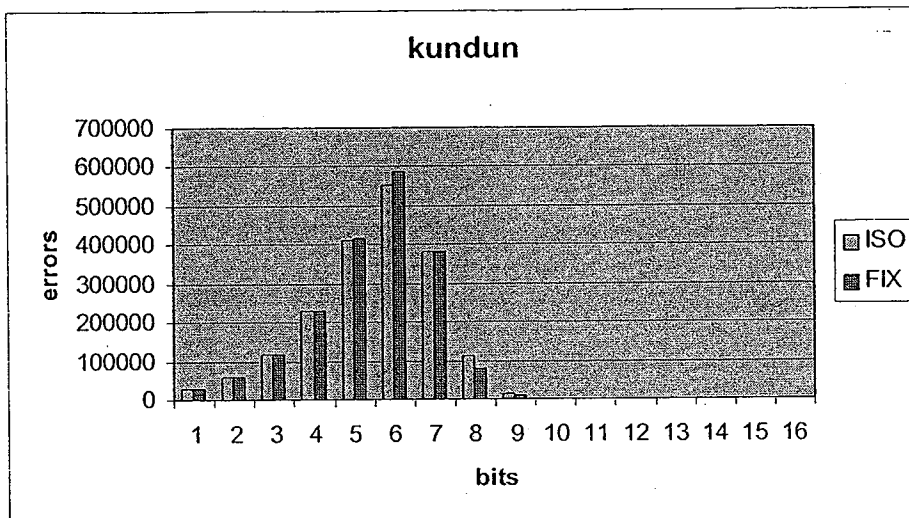
Opera, which is the software equivalent of an ITU-T standard implementing Perceptual Evaluation of Audio Quality (PEAQ), was also used to do verification and testing of the precision study code. Results of opera have been summed up in the following excel sheet.

Opera Results
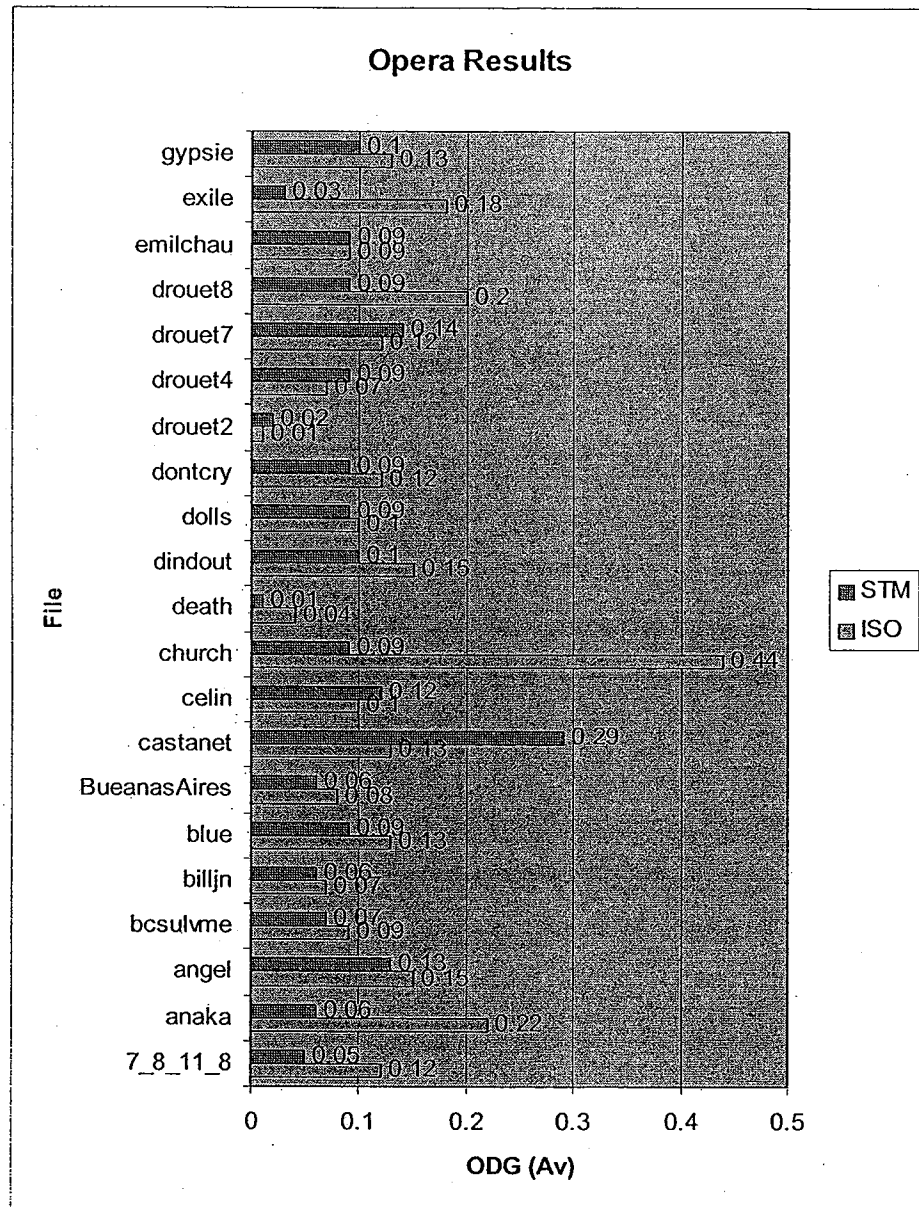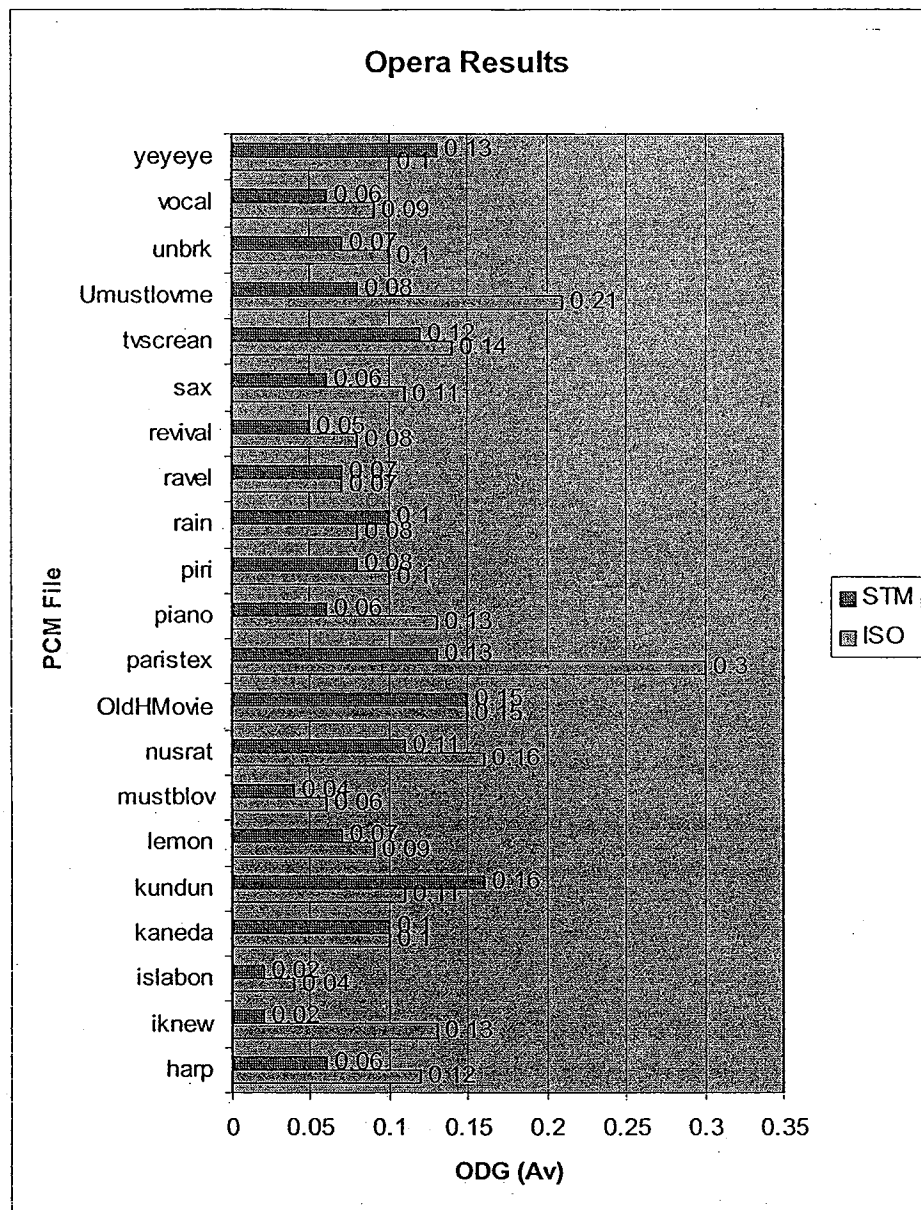
Listening tests are currently undergoing.

4 Reference

[1]International Standard: ISO/IEC 11172-3 "Information technology – Coding of moving pictures and associated audio for digital storage media at up to about 1.5Mbps".